

**IN THE UNITED STATES PATENT & TRADEMARK OFFICE**

Appl. No.: **09/580,163**

Applicant: **Cedara Software Corp.**

Filed: **May 30, 2000**

Title: **ARCHITECTURE FOR AN APPLICATION FRAMEWORK**

Art Unit: **2174**

Examiner: **CHUONG, Truc T.**

Docket No.: **48906-2USPT**

MS Appeal Brief - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**BRIEF ON APPEAL**

**I. INTRODUCTION**

This is an appeal from the Final Office Action dated June 14, 2004 rejecting claims 4-51. A Notice of Appeal from the Primary Examiner to the Board of Patent Appeals and Interferences was filed with the Office, along with a response to the Final Office Action, on October 14, 2004, and the Notice of Appeal was received by the Office on October 18, 2004. This Appeal Brief is being filed in triplicate.

**II. REAL PARTY IN INTEREST**

The real party in interest in the present application is Cedara Software Corp. The assignment was recorded in the United States Patent and Trademark Office at Reel 012028, Frame 0665 in the present application on July 30, 2001.

03/21/2005 MAHMED1 00000014 09580163

01 FC:1402

500.00 OP

### **III. RELATED APPEALS AND INTERFERENCES**

There are no related appeals or interferences in respect of the present application known to the Appellant or Appellant's representative, which will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal.

### **IV. STATUS OF CLAIMS**

In this application, claims 1-3 have been cancelled. Claims 4-51 are pending and are part of the present appeal. Claims 4-51 have been rejected. Please see Appendix A for a complete listing of the claims involved in this appeal.

### **V. STATUS OF AMENDMENTS**

The amendment submitted December 1, 2000 was entered canceling claims 2 and 3 and adding claims 4-51. In response to this amendment, an Office Action dated September 29, 2003 was issued rejecting claims 1 and 4-51. A response was submitted March 29, 2004 which also amended the abstract and canceled claim 1. In response, an Office Action dated June 14, 2004 was issued, finally rejecting claims 4-51. A response to the Final Office Action along with a Notice of Appeal was filed on October 14, 2004. The Notice of Appeal was received by the Office on October 18, 2004. In the response to the Final Office Action, an amendment was entered, amending claims 4 and 17. An advisory action dated December 6, 2004 was then issued, indicating that the request for reconsideration has been considered, that claims 4-51 remain rejected, and that for the purpose of appeal, the proposed amendments would be entered (i.e. those to claims 4 and 17). The rejection appended to the Advisory Action is based on, and is consistent with, the rejection made in the Final Office Action. The rejection of claims 4-51 are the subject of this appeal.

## **VI. SUMMARY OF INVENTION**

The present invention relates to a framework for an application, which seeks to build the application around multiple levels of functionality, and may or may not be visible to the user of the framework. Three levels of functionality are provided as follows.

The first level is a process level, which may also be referred to as a “protocol”, as it encapsulates a set of data plus a definition of the processing steps to be applied. An example of a process instance is a study, which is being viewed in a 2D protocol. This level is managed by the framework and there can be multiple “processes” active at the same time, e.g., the user can be reviewing one study with one process, and at some point, switch to reviewing a study from a different patient (i.e. the framework manages the different process instances that may be active). In a simple case, a system may hide selection at this level, and present a single process to a user. A process includes a set of “activities”, with the possibility of automated control flow from one activity to the next, based on properties of the activities, and external rules.

The second level, is a sub-process level. Although this level is listed as the next level in the hierarchy, it is really just a grouping of activities. This grouping of activities is useful for reuse, convenience of navigation, and grouping activities with shared tools. Generally, a process will make activities visible to the end user, which may even need to have flexibility to switch the order of activities, based on something that is not pre-programmed logic. Thus, the sub-process level may present a user interface for navigating between activities, and for indicating the current position in a process.

The third level, is an activity level. This is the lowest level of the hierarchy that is explicitly supported by the framework (an activity can internally present a deeper model, if required). An activity will have external properties, which are altered as the result of processing, and can then be used to achieve sequencing of activities within a process. It is possible to dynamically (at run time) add or remove activities to a process, so that, for example, a process can provide different functionality based on results of earlier activities. Each activity can be implemented as a separate component.

The present invention also provides a method of monitoring workflow within an application, which has multiple levels of functionality, namely, those described above. The application has a set of activities, which are selectable from a plurality of different sources. The method generally comprises the steps of: selecting a process definition having a set of process steps for processing a data set; selecting the data set associated with the set of activities; initiating the application; navigating between the activities that have been selected from the different sources; and modifying a property contained in the data set for producing an output data set.

## **VII. THE FINAL REJECTION**

In the Final Office Action dated June 14, 2004, the Examiner rejected claims 4-51 under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,144,962 to Weinberg et al. ("Weinberg" hereinafter). In the Advisory Action dated December 6, 2004, the Examiner maintained this rejection after considering the response filed October 14, 2004.

In the Examiner's reasons for rejecting the request for reconsideration, which was appended to the Advisory Action, the Examiner stated *inter alia*:

"The reply does not overcome the final rejection which explained clearly that the URLs or content objects of Weinberg are not only links of Web sites or HTML documents but also image files, Java applets, audio/video files, and Applications (col. 8 lines 32-50); therefore, Weinberg's activities with a set of attributes (a set of attributes, col. 19 lines 23-40) can be used to create, display, edit, and update information including image files, Java applets, audio/video files, and Applications by using the Astra framework of Weinberg as mentioned above. Based on those evidences, the content objects can be controlled/manipulated/navigated using the Astra framework, and the claimed invention and the Astra framework of Weinberg are in the same field (analogous) because of those reasons above and the reference reads on the limitations of claim 4 and others as rejected in the Final Rejection."

The Appellant respectfully disagrees with the Examiner's rejection.

## **VIII. ISSUES**

The issue on appeal in this matter is whether claims 4-51 are patentable over Weinberg, under 35 U.S.C. 102(e).

## **IX. GROUPING OF CLAIMS**

The rejection pertains to claims 4-51, in which claims 4-32 are directed to a framework for monitoring workflow within an application having multiple levels of functionality, and claims 33-51 are directed to a method of the same. Claims 4 and 33 are independent claims, to which the arguments set forth below are directed towards. However, the Appellant considers the claims to be separately patentable and, as such, respectfully submits that the claims do not stand or fall together.

## **X. APPELLANT'S ARGUMENT**

The following arguments are in response to the rejections set forth in the Office Action dated June 14, 2004 and affirmed by the Examiner in the explanation appended to the Advisory Action dated December 6, 2004.

The Examiner has rejected claims 4-51 under 35 U.S.C. 102(e) as being anticipated by Weinberg. The Appellant respectfully traverses this rejection.

### **1) The Present Invention**

As described above in section VI, the present invention relates to an application framework with multiple levels of functionality, which may or may not be visible to the user. The framework has three levels, namely: a process level, a sub-process level, and an activity level. The process level guides the user through a workflow to apply a set of defined process steps associated with a set of activities. The activities comprise their own level of functionality, i.e., the activity level, which modifies a data set through applying the processing of the activities

to create an output data set. The sub-process level aggregates the activities to be applied at the process level.

Accordingly, the framework defines a number of process steps, which are associated with a set of activities selected from the activity level; navigates between the selected activities; and modifies at least one property in the data set according to the process step associated with an activity, to produce an output data set. The activities may be from different sources, and the single application framework defines how the activities from different sources can be mixed to generate various coherent applications. Thus, the individual activities can be outside the framework so that they may be able to utilize the framework when required but may also be used individually as required.

Therefore, each activity has an associated process step, which is applied to a data set, to modify the same, and produce an output data set. The process level selects a set of defined process steps, according to the aggregation of selected activities in the sub-process level, to navigate between the activities, and apply the intended processing to the data set. The multi-level functionality allows, for example, the framework to navigate between the activities, whilst having only the process level or activity level visible to the user. For example, in a medical imaging application, this allows the user to review one study with one process, and at some point, switch to review a study from a different patient (i.e. the framework manages the different process instances that may be active).

## 2) Weinberg

Weinberg teaches a website scanning routine to gather information about the content objects and links of a website via a network connection. Mapping routines use this information to generate, on the computer's display screen, a graphical site map that shows the overall architecture of the website. Weinberg also describes a user interface allowing the user to perform actions such as zooming, panning and filtering. The user can view a comparison map, highlighting changes made to the website, relative to a previous mapping. Weinberg intends to provide a mapping tool for the display and analysis of website content, giving the user the option

to view further information on each data object.

The framework taught by Weinberg relates to a visual mapping architecture, allowing a user to identify certain data objects available on a website, and allows a user to navigate between these data objects. Such an architecture facilitates the management and analysis of websites. The software program employed by the architecture uses conventional webcrawling techniques to gather information about the content objects and links of a website via the network connection. Thus, a visualization of the website is provided, enabling a user to explore the content of the website. The objects may be viewed by zooming, executing a link etc.

However, the content objects are not processed together in a single application, such that the content of a data set is changed, resulting in an output data object. Any manipulation of the data involves display characteristics, which can be changed to suit a particular user's preferences, however these objects are not brought together to generate a single application (i.e. a workflow). Each object is handled and viewed individually. The mapping provides a link to the object, allowing the user to access the object. There exists no workflow to mix the data objects and apply properties thereof to a data set. The data object itself is the data set, and the user is merely interested in searching for and viewing the desired object shown in the visual mapping, or manipulating an attribute associated with that object.

### 3) **Requirements for a Reference to Anticipate**

According to section 2131 of the Manual of Patent Examining Procedure (MPEP), to anticipate a claim, the reference used must teach every element of the claim. The following is taken from appropriate jurisprudence:

“A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.”  
*Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

4) **Does Weinberg Anticipate the Claims of the Present Invention?**

a) **Claim 4**

Claim 4 reads as follows:

4. A framework for monitoring workflow within an application having multiple levels of functionality, said framework capable of combining a plurality of components from different sources, the framework comprising:

- (a) A process level for selecting a set of defined process steps associated with a set of activities to be applied to a data set;
- (b) A sub-process level including an aggregation of selected activities from said set of activities, said sub-process level for facilitating navigation between ones of said selected activities; and
- (c) An activity level including at least one activity from said set of activities; wherein said at least one activity having a property in said data set that is modified as a result of the applied processing of said activity level to produce an output data set.

The Appellant believes that the Examiner has misconstrued the teachings of Weinberg and has therefore improperly equated the teachings of Weinberg (i.e. those used in the Final Rejection), to what is recited in claim 4. The following provides an analysis of each element recited in claim 4 with respect to the teachings of Weinberg.

i) **Preamble**

The preamble of claim 4 sets out the intended use for the framework described therein.

Firstly, the framework is used for monitoring workflow within an application having multiple levels of functionality. In the present application, workflow is described as a set of coordinated activities, which executes several steps, on several objects, wherein the objects may be in different locations. Each of the activities has a different role in the workflow and the



activities may be selected by the user, wherein the workflow coordinates the selected activities according to a set of defined process steps.

Weinberg does not teach a framework for monitoring workflow. Weinberg's Astra framework provides a visual mapping of website content objects and links thereto. The Examiner has stated that Weinberg teaches a framework analogous to that recited in claim 4, however, has provided no guidance as to where in Weinberg these teachings reside. Not only does Weinberg not teach how the Astra framework could be used to monitor a workflow, the framework does not even have the capability of generating a workflow.

Weinberg teaches a mapping, which provides a visual representation of the interconnections and links within a website. Accordingly, the linked objects are stationed in a particular location. There is nothing to even suggest that the objects are to be combined in a workflow. The user may select an object, and even zoom in or run a Java applet associated with the link. This does not, in any way, describe a workflow. The Appellant's believe that the Examiner has erred in overlooking this distinction. The Astra framework taught by Weinberg operates in a completely different way when compared to the framework of claim 4, and operates to achieve a completely different result.

Therefore, Weinberg does not teach a framework for monitoring workflow; and secondly, the Examiner has failed to find (or even suggest for that matter) any teachings in Weinberg that would provide an arguable equivalent.

Secondly, the preamble also states that the framework is capable of combining a plurality of components from different sources. The Appellant's believe that Weinberg's framework does not teach such functionality. The Appellant wishes to note that for this limitation, the Examiner has also failed to cite any teachings in Weinberg that provide an equivalent.

Weinberg teaches a mapping of website objects. Each of the objects may be accessed through a link displayed in the mapping for each object. Although the objects are "linked" together, i.e., to graphically display the mapping, they are not "combined" from different sources

to generate a workflow. Since Weinberg does not teach a workflow of activities, the “linking” of objects in a mapping cannot be reasonably equated to “combining” the objects, since Weinberg does not teach coordinating the objects displayed in the mapping of the Astra framework in a single application (i.e. a workflow).

Therefore, Weinberg does not teach a framework that is capable of combining a plurality of components from different sources.

ii) Element (a)

Element (a), recited in claim 4, describes a process level. The process level is used for selecting a set of defined process steps associated with a set of activities to be applied to a data set. The set of defined process steps dictates which activities are to be applied to the data set. This, in a general sense, describes a workflow. The workflow includes a set of activities, which it coordinates, to manipulate the data set. The Examiner has cited col. 8 lines 9-20 and lines 32-50, figures 1-3, and col. 19 lines 23-40; and believes that these citations anticipate element (a). The Appellant respectfully disagrees.

Firstly, if one were to equate the Astra framework taught by Weinberg with the process level of claim 4, it would be required that the Astra framework be used for selecting a set of defined process steps. Although the Astra framework allows a user to select an object displayed in the graphical mapping, Weinberg does not teach selecting a set of defined process steps. Each object displayed in the Astra framework can be selected, and the visual characteristics manipulated. However, there is nothing in the Astra framework that teaches selecting a set of defined process steps. In fact, doing so would be contrary to the described purpose of the Astra framework, namely, to allow a user to selectively access a desired object, one at a time, through its associated link. The steps taken by a user are not part of a set of defined process steps. Weinberg does not even suggest how a user would use the framework in a defined way, only that the mapping exists, and that the user may use the mapping to link to an object. The purpose of the Astra framework is to allow a user to access and/or manipulate objects one at a time.

Secondly, since Weinberg does not teach a set of defined process steps, there is also no teaching of a set of defined process steps which are associated with a set of activities. The Examiner believes that since the objects in the graphical display map have display attributes, the Astra framework anticipates element (a). The Examiner has failed to consider every limitation recited in element (a). Even if the objects in the display mapping of the Astra framework were equivalent to data sets, and the display attributes associated with the objects were equivalent to activities, Weinberg still does not teach selecting a set of defined process steps associated with a set of these attributes to be applied to an object. The purpose of allowing a user to manipulate the display attributes is to provide the capability of selecting user-defined attributes and there is no teaching of a set of defined steps that apply a selected set of attributes to an object.

Therefore, Weinberg does not teach a set of defined process steps which are associated with a set of activities.

iii) Element (b)

Element (b), recited in claim 4, describes a sub-process level. The sub-process level represents an aggregation of selected activities from the set of activities. The sub-process level is used for facilitating navigation between the selected activities. In other words, the sub-process level coordinates the workflow, and moves from one activity to the next, according to the set of defined process steps selected in the process level.

The Examiner has primarily relied on the same text from Weinberg regarding element (a) in finding that element (b) is anticipated by the Astra framework. According to the analysis above regarding element (a), it appears that the Examiner believes the activities recited in claim 4 are equivalent to the display attributes described by Weinberg. The display attributes include a number of display properties that may be changed by the user. Weinberg does not teach an aggregation of a selected number of display attributes. Similarly, Weinberg does not teach facilitating navigation between the selected activities. The Appellants believe that the Examiner has again failed to consider every limitation recited in claim 4. Specifically, Weinberg does not teach the functionality described in element (b), let alone a sub-process level (i.e. a sub-level to

the Astra framework) that includes an aggregation of selected activities.

If the Examiner wishes to equate the collection of display attributes in Weinberg, to an aggregation of selected activities as recited in claim 4, then it is unclear what constitutes the group of activities from which this aggregation is selected. On the other hand, if the Examiner wishes to equate the collection of display attributes in Weinberg, to the entire set of activities in claim 4, then what constitutes the aggregation of selected activities? It appears that the Examiner has again overlooked the specific limitations recited in the claim. Furthermore, the navigation between display attributes is not done on a sub-process level in the Astra framework. The user manipulates the display attributes directly at a level that the Examiner believes to be equivalent to the process level of claim 4, and not a sub-process level.

Accordingly, since the Astra framework does not monitor or facilitate workflow, there is no need for multiple levels of functionality. The display attributes may be manipulated, but this is done on an attribute-by-attribute basis and even if one were to consider the display attributes to be equivalent to the activities recited in claim 4, there is no coordination of attributes according to a set of defined process steps.

The Examiner appears to have used *ex post facto* analysis in concluding that Weinberg's framework operates in an analogous manner to what is recited in claim 4. In fact, there is no teaching therein that would enable a person skilled in the art to use the framework in an analogous manner, or even whether the Astra framework is capable of performing the functionality recited in claim 4.

Therefore, Weinberg does not teach a sub-process level, which includes an aggregation of selected activities.

iv) Element (c)

Element (c), recited in claim 4, describes an activity level. The activity level includes at least one activity, selected from the set of activities. The activities have properties in the data set

that are modified as a result of the applied processing of the activity level, to produce an output data set.

The Examiner has clearly equated the set of attributes associated with each of the objects in the display map with the activities recited in claim 4. Since the attributes relate to display properties that are applied to the objects, the Examiner believes that the objects are equivalent to the data set described in claim 4. Using this interpretation, Weinberg could, for argument's sake, be said to include an activity level. However, if this is the case, then the activity level is equal to what the Examiner also believes is the sub-process level and the process level. Therefore, according to the Examiner's interpretation, the Astra framework does not comprise multiple levels of functionality but only one level of functionality. Furthermore, even if one were to assume this is true, the Astra framework still does not teach monitoring workflow, teach a sub-process level, or teach a set of defined process steps. Thus, for the Examiner's interpretation to hold, one would have to look at element (c) in isolation, without any regard to the other elements of claim 4.

Accordingly, if one were to find that the teachings of Weinberg are equivalent to element (c), one would have to disregard the other elements recited in claim 4. Therefore, claim 4 would not be anticipated for at least that reason.

v) Summary of the Patentability of Claim 4

Accordingly, the Appellants believe that Weinberg does not anticipate claim 4. Specifically, the Astra framework described by Weinberg does not monitor a workflow within an application, does not combine a plurality of components from different sources, does not select a set of defined process steps associated with a set of activities, does not include an aggregation of selected activities from a set of activities, and does not facilitate navigation between the members of the aggregation. Furthermore, even if one were to find element (c) equivalent to the Astra framework, then the Astra framework would fail to teach the other limitations of claim 4.

Therefore, it is believed that claim 4 clearly and patentably distinguishes over Weinberg

and is in condition for allowance. Claims 5-32 are either directly or indirectly dependent on claim 4 and therefore are also believed to distinguish over Weinberg.

b) Claim 33

Claim 33 is directed towards a method for monitoring workflow, having the same features as claim 4. Although similar arguments can be applied in favour of claim 33, a few observations will be made.

Claim 33 reads as follows:

33. A method of monitoring a workflow within an application having multiple levels of functionality, said application having a set of activities selectable from a plurality of different sources, the method comprising the steps of:

- (a) selecting a process definition having a set of process steps for processing a data set;
- (b) selecting said data set associated with said set of activities;
- (c) initiating said application;
- (d) navigating between ones of activities selected from said set of activities; and
- (e) modifying a property contained in said data set for producing an output data set.

Claim 33 is directed towards a method which uses a framework such as the one recited in claim 4. The method of claim 33 is directed to monitoring a workflow. According to the above, Weinberg does not teach monitoring a workflow, nor even teach of the possibility of generating a workflow. Therefore, for at least that reason, Weinberg does not anticipate claim 33.

Furthermore, step (a) recited above, requires that a process definition be selected. Weinberg does not teach such a step. Similar arguments regarding the remaining steps can be made using the arguments presented in favour of the patentability of claim 4.

Accordingly, Weinberg does not teach all the steps recited in claim 33, and thus does not anticipate claim 33. Therefore, claim 33 is believed to distinguish over Weinberg and as such is believed to be in condition for allowance. Claims 34-51 are either directly or indirectly dependent on claim 33 and, as such, are also believed to distinguish over Weinberg.

c) Conclusions on the Patentability of the Claims over Weinberg

In view of the foregoing, and in accordance with section 2131 of MPEP, the Appellant believes that Weinberg does not teach every element of independent claims 4 and 33, nor those claims dependent thereon. Therefore, the Appellants believe that claims 4-51 clearly and patentably distinguish over Weinberg, and as such, the claims of the present application are believed to comply with 35 U.S.C. 102(e), and are in condition for allowance.

**XI. CONCLUSION**

In summary, the Appellant respectfully submits that the Examiner has erred in his rejection of claims 4-51 as being anticipated by Weinberg. Weinberg does not teach all of the elements of any of the claims, as required for a reference to anticipate a claim. Therefore, the requirements of section 2131 of MPEP have not been met, and the claims of the present application define patentable subject matter.

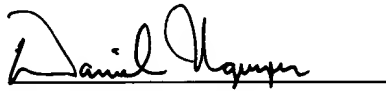
Not only does Weinberg fail to teach all of the elements of the claims, the Examiner has failed to make reference to any teachings in Weinberg that describe certain ones of the limitations of claim 4, namely, monitoring a workflow, or selecting a set of defined process steps.

Accordingly, the Appellants believe that the Examiner has overlooked several important distinctions between the claims of the present application and Weinberg, and has thus erred in finding that Weinberg anticipates the claims. Therefore, the Appellant believes that claims 4-51 clearly and patentably distinguish over Weinberg, and are in condition for allowance.

The Appellant respectfully requests that this honourable Board of Patent Appeals and Interferences reverse the Examiner's decision in this case and indicate the allowability of claims 4-51 in this application.

Dated: March 17, 2005

Respectfully submitted,

A handwritten signature in black ink, appearing to read "Daniel G. Nguyen", written over a horizontal line.

Daniel G. Nguyen  
Registration No.: 42,933  
JENKENS & GILCHRIST, A PROFESSIONAL  
CORPORATION  
5 Houston Center  
1401 McKinney, Suite 2600  
Houston, Texas 77010  
(713) 951-3300  
Attorneys for Applicant



## **APPENDIX A**

Listing of claims involved in this appeal:

4. A framework for monitoring workflow within an application having multiple levels of functionality, said framework capable of combining a plurality of components from different sources, the framework comprising:

- (a) A process level for selecting a set of defined process steps associated with a set of activities to be applied to a data set;
- (b) A sub-process level including an aggregation of selected activities from said set of activities, said sub-process level for facilitating navigation between ones of said selected activities; and
- (c) An activity level including at least one activity from said set of activities; wherein said at least one activity having a property in said data set that is modified as a result of the applied processing of said activity level to produce an output data set.

5. A framework according to claim 4 further comprising a user interface for facilitating interaction between a user and said application.

6. A framework according to claim 5, wherein the levels are assignable to distinct regions of said user interface.

7. A framework according to claim 6, wherein said activity level further supports re-use of a previous activity over a current activity, said activity selected from said aggregation selected activities.

8. A framework according to claim 5, wherein said user interface includes a screen for providing a display of images.

9. A framework according to claim 8, wherein a current activity being processed from said set of activities is assigned to a work area of said screen, said work area having a substantial portion of the screen surface area.
10. A framework according to claim 9, wherein said framework monitors ownership of said work area by said current activity.
11. A framework according to claim 6, wherein said user interface facilitates multiple activities that are processable concurrently.
12. A framework according to claim 4, wherein said sub-process level facilitates a dynamic ordering of said selected activities by said user.
13. A framework according to claim 4, wherein said process level automates a control flow between said selected activities in said set of activities based on a rule set or an activity property set.
14. A framework according to claim 4, wherein at least two of said different sources have different formats.
15. A framework according to claim 4, wherein said process level monitors functionality of a current activity based on said output data set obtained from a previous activity.
16. A framework according to claim 4, wherein said process level includes a data selector for selecting said data set.
17. A framework according to claim 16, wherein said process level further includes a process selector for selecting said set of defined process steps compatible with said data set.
18. A framework according to claim 4, wherein said process level facilitates selection between active activities by a user.

19. A framework according to claim 4 further comprising a tool level for setting a parameter of said activity level, said parameter for updating an operational behavior of said activity level.
20. A framework according to claim 19, wherein said tool level is assignable to a distinct region of a user interface, said user interface for facilitating interaction between a user and said application.
21. A framework according to claim 20, wherein said framework coordinates installation of a tool in the tool level region of said interface, said tool requested by said activity level.
22. A framework according to claim 19, wherein said tool level includes a tool navigator for facilitating selection of a tool by said user.
23. A framework according to claim 19, wherein multiple tool levels are supported by said framework.
24. A framework according to claim 9, wherein a content of said work area contains shared properties stored in a shared data context.
25. A framework according to claim 24, wherein said shared data context is accessible by cooperating ones of said selected activities for sharing information.
26. A framework according to claim 24, wherein said data set and said set of process steps form a basis of said shared data context.
27. A framework according to claim 24, wherein the content of said shared data context accessible by said user for verifying that required data for said selected activities is present.
28. A framework according to claim 4, wherein said framework restricts access by said user of selected ones of the levels.
29. A framework according to claim 4 further including a module for interfacing said application to a database library.

30. A framework according to claim 29, wherein said database library includes data selected from the group comprising process definitions, sub-process descriptions, and activity information.

31. A framework according to claim 29, wherein said data set is external to said framework with an interface to said data set provided by said module.

32. A framework according to claim 19, wherein said framework restricts access by said user of selected ones of the levels.

33. A method of monitoring a work flow within an application having multiple levels of functionality, said application having a set of activities selectable from a plurality of different sources, the method comprising the steps of:

- (a) selecting a process definition having a set of process steps for processing a data set;
- (b) selecting said data set associated with said set of activities;
- (c) initiating said application;
- (d) navigating between ones of activities selected from said set of activities; and
- (e) modifying a property contained in said data set for producing an output data set.

34. A method according to claim 33 further comprising the step of facilitating interaction between a user and said application by a user interface.

35. A method according to claim 34 further comprising the step of assigning at least some of said multiple levels of functionality to distinct regions of said user interface.

36. A method according to claim 35, wherein said user interface includes a screen providing a display of images.

37. A method according to claim 36 further comprising the step of assigning a current activity selected from said set of activities to a work area of said screen, said work area having a substantial portion of the screen surface area.
38. A method according to claim 37 further comprising the step of monitoring ownership of said work area by said current activity.
39. A method according to claim 35 further comprising the step of processing at least two process definitions concurrently.
40. A method according to claim 33 further comprising the step of dynamically monitoring of an execution order of said set of activities by a user of said application.
41. A method according to claim 40 further comprising the step of automating a control flow between selected activities in said set of activities based on a rule set or an activity property set.
42. A method according to claim 33 further comprising the step of monitoring an operational functionality of said set of activities based on said output data set obtained from a previous activity.
43. A method according to claim 39 further comprising the step of selecting between active activities for assignment to a work area of said user interface.
44. A method according to claim 43, wherein said application supports a reuse of a previous activity over a current activity, said previous activity selected from said set of activities.
45. A method according to claim 33 further comprising the step of setting a parameter of said set of activities by a tool, said parameter for updating an operational behavior of said set of activities.

46. A method according to claim 45 further comprising the step of assigning said tool to a distinct region of a user interface, said user interface for facilitating interaction between said user and said application.
47. A method according to claim 46 further comprising the step of installing a tool in the tool region of said user interface, said tool requested by said process definition.
48. A method according to claim 34 further comprising the step of sharing properties of a content of a work area of said user interface in a shared data context.
49. A method according to claim 48 further comprising the step of accessing said shared data context by cooperating ones of said set of activities for sharing information purposes.
50. A method according to claim 35 further comprising the step of monitoring a level of information displayed in said distinct regions of said user interface.
51. A method according to claim 35 comprising the step of managing said distinct regions of said user interface by a current activity.